



# Virtual environment trajectory analysis: a basis for navigational assistance and scene adaptivity

Corina Sas<sup>a</sup>, Gregory O'Hare<sup>a,\*</sup>, Ronan Reilly<sup>b</sup>

<sup>a</sup> Department of Computer Science, University College Dublin, Dublin, Ireland

<sup>b</sup> Department of Computer Science, National University of Ireland Maynooth, Ireland

Available online 15 July 2004

## Abstract

This paper describes the analysis and clustering of motion trajectories obtained while users navigate within a virtual environment (VE). It presents a neural network simulation that produces a set of five clusters which help to differentiate users on the basis of efficient and inefficient navigational strategies. The accuracy of classification carried out with a self-organising map algorithm was tested and improved to in excess of 85% by using learning vector quantisation. This paper considers how such user classifications could be utilised in the delivery of intelligent navigational support and the dynamic reconfiguration of scenes within such VEs. We explore how such intelligent assistance and system adaptivity could be delivered within a Multi-Agent Systems (MAS) context.

© 2004 Elsevier B.V. All rights reserved.

**Keywords:** Spatial behaviour; Trajectory classification; Adaptive VEs

## 1. Introduction

This study investigates how the analysis of users' movement patterns, in particular trajectory classification, can be harnessed for designing adaptive VEs. Such VEs are designed to provide intelligent navigational support, particularly to low spatial users. The motion trajectories were recorded while a set of subjects carried out spatial tasks within a VE.

Prior studies in the area of spatial cognition were concerned with testing hypotheses concerning the im-

pact of a variety of factors on spatial knowledge acquisition [5]. However, none of these tried to investigate holistically the motion trajectories themselves. Trajectory analysis can support the extraction of valuable information concerning those navigational rules which users employ in accomplishing spatial tasks. In addition, when this analysis is performed in the light of some performance criterion (e.g., time required to perform a search task) it could provide valuable insights into discriminating efficient and inefficient navigational strategies and clustering users accordingly.

Attempts to cluster trajectories have been carried out primarily in the area of visual surveillance, especially novelty detection, with the purpose of identifying

\* Corresponding author. Tel.: +353 1 716 2472; fax: +353 1 269 7262.

E-mail address: gregory.ohare@ucd.ie (G. O'Hare).

suspicious behaviour of pedestrians within an outdoor open area [7,20]. This goal is directly linked to the idea of automatic surveillance, which would allow the replacement of the human operator. In their study, Owens and Hunter have shown that the Self-Organising feature Map (SOM) neural network could successfully be employed to perform trajectory analysis by both identifying the characteristics of normal trajectories and detecting novel trajectories [20].

However, trajectory analysis performed on a spatial cognition task represents a novel approach. The objective of this study involves identifying the efficient and inefficient movement trajectories and their associated characteristics. Efficient versus inefficient is determined in the light of users' performances.

Without underestimating the role of traditional clustering methods, we propose the use of Artificial Neural Networks (ANN) as an alternative tool for trajectory classification. Neural networks provide a very powerful toolbox for modelling complex non-linear processes in high dimensionalities [14]. ANNs have many advantages over traditional representational models, particularly given their distributed representations, parallel processing, robustness to noise or degradation and biological plausibility [9]. We consider that at least part of these strengths can be harnessed to model user's navigational behaviour.

The paper is organised as follows. The next section introduces both self-organising maps and Learning Vector Quantisation (LVQ), and advocates using them to perform clustering analysis. This is followed by a section describing the study design, in terms of apparatus, procedure, sample and data collection. We subsequently present, a detailed description of data analysis and study results which led to five trajectory clusters. Having successfully identified these clusters, the subsequent section tries to exploit this classification, as an initial step in providing intelligent navigational support within VEs. Finally, the benefits of the study outcomes are summarised, and some promising research trends are suggested.

## 2. Cluster analysis performed by artificial neural networks

The main goal of cluster analysis is data reduction, by subdividing a set of objects into a hierarchical ar-

rangement of homogeneous subgroups. A significant outcome is reduced complexity with a minimal loss of information, which fosters a better understanding of the analysed data [15]. An important aspect of any clustering method is the minimisation of classification errors. As Kaski [10] pointed out, one problem usually associated with clustering methods is the interpretation of clusters. Due to their ability to extract patterns and to visualise complex data in a two-dimensional form [10], SOM is used to perform the trajectory cluster analysis. Like many other clustering techniques, SOM reduces representations to the most relevant facts, with minimal loss of knowledge about their interrelationships [10].

The SOM is a neural network algorithm, which exhibits several distinct advantages over other clustering techniques [10,20]. The mapping from a high dimensional data space onto a two-dimensional output map is effectively used to visualise ordering relations of input data. Reducing the amount of data allows comprehensible cluster identification and interpretation, which is an inherently difficult task in the case of traditional clustering methods [10]. As with any other ANNs, SOM has considerable potential to generalise. Thus once it is trained, a SOM is able to classify new data against the set of clusters previously identified. All these features highly commend SOM as a basis for on-line automatic extraction of trajectory clusters.

In addition to the basic features of SOM we outline SOM and learning vector quantisation as unsupervised and supervised learning processes, respectively. LVQ constitutes a supervised learning algorithm related to SOM. The SOM and LVQ algorithms were developed by Kohonen et al. and implemented by his team from Helsinki University of Technology, in the form of SOM\_pak [12] and LVQ\_pak [13]. These comprehensive software package are available online and were used within this study. SOM is based on an unsupervised learning process, allowing both the cluster identification within the input data and the mapping of an unknown—not previously seen—data vector with one of the clusters. This process is carried out without any prior knowledge regarding number and content of the clusters to be obtained [10]. When a set of already clustered input data is available, a supervised learning process can be employed to identify to which class an unknown data vector belongs.

### 2.1. Self-organising maps

A basic SOM consists of an input layer, an output map and a matrix of connections between each output unit and all the input units. The input is usually represented by a multidimensional vector with each unit coding the value from one dimension. Every node from the two-dimensional output layer is associated with a so-called reference vector ( $m_i$ ), which consists of a set of weights from each input node to the specified output node. Each input vector is compared with all the reference vectors and the location of the best match according to some metric, usually the smallest of the Euclidean distances, is defined as the winner. Around the maximally responding unit, a topological neighbourhood is defined and the weights of all units included in this neighbourhood are adjusted, according to Eq. (1), where  $m_i$  is the weight at time  $(t + 1)$  and  $\eta$  the learning rate:

$$m_i(t + 1) = m_i(t) + \eta[x(t) - m_i(t)]. \quad (1)$$

The topological neighbourhood should be initially quite large, enabling a global order of the map, while in the subsequent stages its values are decreased as a function of time. Accordingly, the learning rate varies in time from an initial value close to unity, to small values over a long time interval. Training is performed during two phases: an ordering phase during which the reference vectors of the map units are ordered (neurons in different areas of the network learn to correspond to coarse clusters in the data), and a much longer fine-tuning phase during which the reference vectors in each unit converge to their correct values (neurons adjust to reflect fine distinctions).

The learning process consists of a “winner-takes-all” strategy, where the nodes in the output map compete with each other to represent the input vectors. For this reason, the output layer is also called the competitive layer. Competitive learning is an adaptive process, through which the neurons from the output layer become slowly sensitive to the input data, learning to represent better different types of inputs.

As Kohonen et al. [13] pointed out, a significant property of SOM is the tendency to preserve continuity in terms of mapping similar inputs to neighbouring map locations. This is influenced by the weight vectors trying to represent the density function of the input

vectors. As a result of these antagonistic tendencies, the distribution of reference vectors is rather smooth, given the search for an optimal orientation and form to match those of the input vector density. In addition, the greater the variance between the input vector features, the better their representation on the output map. It would be expected that these features correspond to the most important dimensions of the inputs.

### 2.2. Learning vector quantisation

LVQ consists of an input layer comprising multidimensional vectors described by their features and an output layer whose neurons correspond to the predefined classes. There is also a matrix of connections between each output unit and all the input units, consisting of weight vectors. Since each weight vector corresponds to a class, they are considered as labelled. The basic idea is that input vectors belonging to the same class will cluster in data space, in the form of a normal distribution around a prototype vector. Classifying an input vector consists of computing the Euclidean distance between the considered input vector and all the weight vectors, followed by assignment to the class associated with a weight vector for which the Euclidean distance is a minimum [13].

During training, an adaptive process occurs with respect to the closest weight vector, also called the winning neuron. When both the input vector and the weight vector belong to the same class, meaning that the input vector was correctly classified, the weight vector is modified in order to become a better approximation of the input vector. However, when the input vector is incorrectly classified, the weight vector is adjusted in a manner such that it increases the distance of the input vector (since they belong to different classes).

While the SOM algorithm strives to approximate the weight vectors to the input vectors, LVQ tries to lead to weights that effectively represent each class. The process of adjusting the weights, without respect to any topological neighbourhood differentiates LVQ from SOM. The performance of LVQ can be increased by initialising the codebook vectors with those values obtained as a result of training the SOM [10]. Variants of SOM have been successfully applied to a large number of domains, ranging from monitoring and control of industrial tasks, to robot navigation, from data processing to machine vision, from image analysis to



Fig. 1. Virtual conference room.

novelty detection [10]. However, their adoption within the frame of spatial cognition in VEs constitutes a novel approach.

### 3. Study design

VEs have become a rich and fertile arena in the investigation of spatial knowledge. Within the VEs, the user set of actions is restricted, consisting mainly of navigation and locomotion, object selection, manipulation, modification and query [6]. VEs serve as an efficient and powerful tool which facilitate accurate spatio-temporal recording of users' trajectory within the virtual space [1]. Attempts to understand spatial behaviour in both real and artificial worlds have hitherto been primarily concerned with highlighting the symbolic representation of spatial knowledge.

In this study we utilise ECHOES [3,18,19] as an experimental test-bed. ECHOES is a virtual reality system which offers a small-scale world, dense, static and with a consistent structure. Adopting a physical world metaphor, the ECHOES environment comprises a virtual multi-story building, with each of the levels containing several rooms: conference room (Fig. 1), library (Fig. 2), lobby, etc. Users can navigate from level to level using a virtual elevator. The rooms are furnished and associated with each room there is a cohesive set of functions provided to the user. These features enable ECHOES to offer an intuitive navigational model.



Fig. 2. Virtual library.

A sample of 30 students were asked to perform two tasks within the VE. The first, an exploratory task, provided the primary data for the trajectory classification, while the second, a searching task, offered a basis for assessing the quality of exploration and the efficiency of the exploratory strategy. The time needed to search for a particular room acts as a performance indicator of the level of spatial knowledge acquired within the VE. Thus, the shorter the search time, the better the spatial knowledge [21]. According to the time required for the search task, users have been identified as low *spatial users*, when they needed significantly longer time to find the library, or *high spatial users* who found the library straight away. Each of these two groups is represented by about 20% of users participating in this study.

A comprehensive set of data was recorded throughout the experiment. This was achieved by way of intelligent agent technology. A listener agent was encoded with certain event activation thresholds and subsequently commitments were adopted to future directed actions, namely to record. This agent was delivered as a strong agent and realised within Agent Factory a rapid prototyping environment for agents developed in part by one of the authors [2,17]. Specifically movements greater than half a virtual metre, and each turn greater than 30°, were recorded. This was achieved by the inclusion of a rich set of virtual sensors or actuators attached to the agent together with an odometer and rotational event listener.

### 4. Data analysis and results

By providing a rich set of primary data, trajectory analysis can support the extraction of valuable information regarding the rules users employ in accomplishing spatial tasks. Moreover, when this analysis is performed in the light of some performance criterion (e.g., time required to perform a search task) it can provide valuable insights into discriminating efficient and inefficient navigational strategies and clustering the users accordingly.

Trajectory classification provides the benefits of reducing the huge amount of information stored in raw data and once a typology has been created it can be used to assess any new trajectory by comparing and assigning it to the appropriate class. On-line trajectory

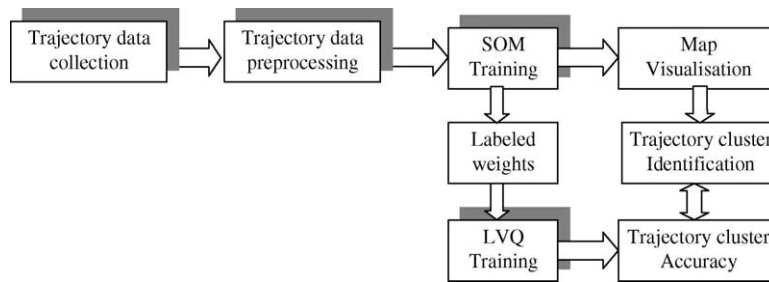


Fig. 3. Modular system for on-line trajectory classification.

classification would allow the identification of user's in terms of good or poor performers of spatial tasks. This identification could represent an essential initial step in designing the adaptive VE. Thus, the VE could be dynamically reconfigured in order to enable poor users to learn the efficient navigation procedures, while for good performers, it can be redesigned in order to challenge users' spatial skills.

The use of SOM [12] and LVQ [13] in performing trajectory cluster analysis requires several steps: data collection, construction and normalisation of the data set, unsupervised training, visualisation of the resulting map, cluster identification, production of a set of trained labelled codebook vectors to be used in supervised training and finally the measurement of the classification accuracy. We have developed a modular architecture which encompasses these steps supporting the automatic and seamless movement from one stage to the next. This architecture is depicted in Fig. 3.

#### 4.1. Collecting data

The data collection module is based on the listener agent previously developed by O'Hare et al. [18,19]. The listener agent constantly harvests information about user behaviour within the virtual world. The data captured when the user interacts with the ECHOES virtual space contains details of navigation paths through the world together with the time spent in different rooms.

#### 4.2. Pre-processing data

Data pre-processing involves the transformation of the raw data into a suitable form to be fed into

the SOM. Each trajectory is represented within the raw data as a multivariate time series. However, we were interested in testing whether the possibility of commissioning a static representation of trajectory is sufficient to perform a classification. Thus it was necessary that the reduction of the raw data should be accomplished by preserving their significant features. We chose to represent each trajectory by the degree of occupancy of a predefined set of spatial locations.

For the SOM analysis, we overlaid the virtual space with a grid composed of 28 squares of  $4 \times 4$  virtual metres. Each trajectory was converted to a succession of locations upon the grid. Subsequently it was necessary to map each trajectory into a sequence of 28 neurons (one for each location), according to Eq. (2), where NV is the input node value and LOC is location occupancy expressed as how many times the user revisited that location:

$$NV = \log_{10} 9LOC + 1. \quad (2)$$

The above transformation permits a clear differentiation between non-visited ( $NV = 0$ ) and visited locations. In the later case the NV is within the range 1–2, 1 for only one visit and 2 for 11 visits, 11 being the maximum number of times for revisiting a location. In addition to this encoding which features space coverage, the trajectories were augmented by characterising the amount and size of users' rotations. We considered trajectories involving rotation angles in excess of  $90^\circ$  to represent an interesting feature. If a trajectory has more than 10% of the rotation angles equal or greater than  $90^\circ$ , the 29th node of the input vector was set to 3, otherwise it was set on 0.



#### 4.3. Training SOM

Once the data is pre-processed, we randomly divided it into two equal subsets, one for training and the other one for the testing. Each set consisted of 63 vectors, comprising encoded trajectories covered by the users on each level. A SOM of  $16 \times 12$  neurons was used to perform a topology-preserving mapping.

The first phase of training was carried out for 1000 epochs, a radius of 16 and with a learning rate of 0.8, while the second phase was 120,000 epochs, with a learning rate of 0.01 and a radius of 2. The random seed was 275 identified by using the *vfind* program. These parameters were retained, after we tried more than 50 trainings, with different architectures and learning rates, since they led to the smallest *quantisation error* for the testing set (1.97), while for the training set it was 0.35. Quantisation error represents the norm of difference of an input vector from the closest reference vector [12].

#### 4.4. Map visualisation

The resulting organisation of the map, shown in Fig. 4 shows five clusters of users, where clustering is on the basis of their navigational pattern within the VE.

Numbers which were associated with the winner neurons within each cluster were replaced by the original corresponding trajectory. SOM provides an additional visualisation benefit depicting cluster boundaries with darker shades of grey, since they represent larger distances between adjacent neurons [11].

#### 4.5. Cluster identification

Training the SOM led to the identification of five clusters [23]. In order to assist identification we placed the associated cluster number on the SOM map for example cluster number 1 within the area designated by number 1, located in the middle of the lower half of the map. Cluster 1 groups trajectories limited to the lower half, for example two rooms of the spatial layout. These trajectories are generally circular. Some of them are smooth while others contain sharper angles.

Cluster 2 located on the top left quadrant of the map comprises trajectories, which contain lots of turns and usually intersect themselves. These trajectories are completely different from any other group, containing straight lines joined at sharp angles. They do not facilitate an efficient coverage of the space and are more likely to induce disorientation and accordingly the level of spatial knowledge which can be acquired through

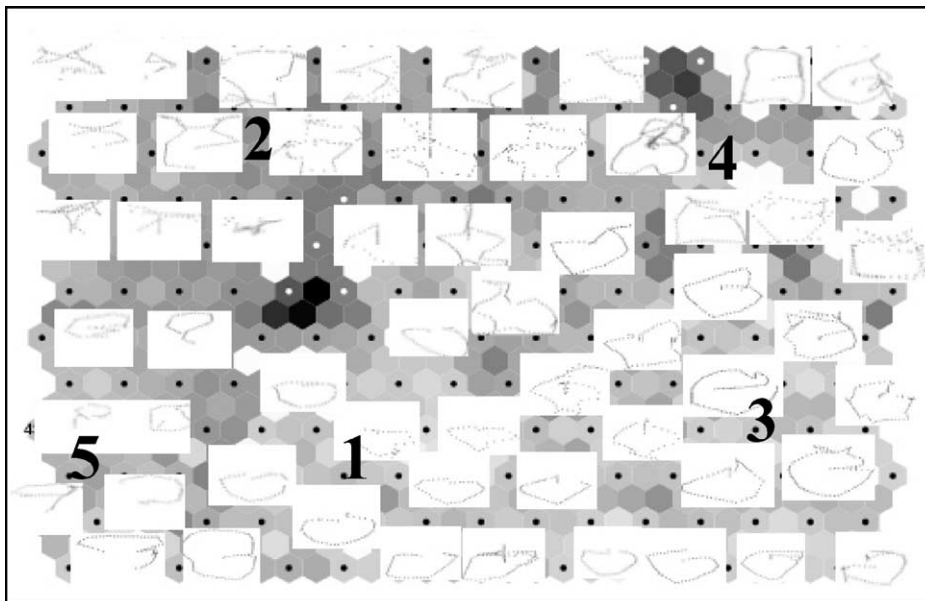


Fig. 4. SOM map obtained from the training set.

them is somewhat limited. As it can be seen, there are two sub-clusters that can be identified within this class, those whose main distinction resides in the coverage of the space. For some of the trajectories, the coverage is restricted to only one room within the space, while the rest of them provide a larger coverage, which rarely become circular. Within such trajectories there is an erratic nature and the user seems anxious to explore the space, for example he/she moves in the same area or covers a larger space. In this case, however, it is likely that their return to the starting point is achieved through a non-circular path.

Cluster 3 located on the right part of the second half of the map, consists of very smooth circular trajectories, which have at least one direction towards the centre of the spatial layout. Cluster 4 comprises longer trajectories, which cover most of the spatial layout. They demonstrate the “going around the edge” feature, in a much more pronounced manner than other clusters, with the exception of Cluster 2. Cluster 5 presents circular trajectories performed within the first half of the spatial layout.

Each set of trajectories, with the exception of Cluster 2, proves beneficial along the temporal dimension. Carefully selected and ordered, they enable users to acquire particular spatial knowledge, with a minimum investment of resources.

Previous work in classifying trajectories, performed only on the basis of locations, led to a more detailed classification [22]. However, since the purpose of this analysis is to discriminate between users employing efficient strategies and those navigating through a set of inefficient strategies, we do have to take the rotation angle into account. This leads to a more detailed representation of Cluster 2.

#### 4.6. Training LVQ

Once the SOM was trained, the codebook vectors could be used in initialising the weights for the LVQ algorithm. This did indeed lead to increased classification accuracy increasing from 72% obtained using random initialisation to 87%. In other words, each trajectory from the testing set was correctly classified by the LVQ with 87% accuracy. Within each class, the classification accuracy is slightly different—Cluster 1: 86%, Cluster 2: 100%, Cluster 3: 63%, Cluster 4: 87% and Cluster 5: 100%. As can be seen, the trajectories

belonging to Cluster 2, which require special attention are correctly classified in each case. This is an important outcome, which supports the original goal of our study, namely to discriminate users in terms of good and poor performers on the spatial tasks. This finding should also be emphasised in the light of the fact that, more than 50% of trajectories comprising Cluster 2, are covered by the subjects with worst performance in the searching tasks. In other words, Cluster 2 provides an interesting benefit by binding users’ spatial behaviours, for example movement paths, with their spatial abilities, for example performance on search tasks.

### 5. Intelligent navigation and scene adaptivity

Having successfully identified user clusters that correspond to broad categories of spatial navigational competencies we sought to harness these classifications in order to underpin intelligent navigational support and scene adaptivity within VEs. Based upon our work we are able to effectively monitor user interactions and patterns of navigation and exploration through our listener agent and subsequently dynamically able to match such behaviour patterns against the trajectory clusters. Upon identification of the cluster to which an individual user belongs we can potentially make inferences based upon their competencies as to the level of navigational support they might need and the likelihood of success in exploration and retrieval tasks.

One cannot discuss intelligent navigation support, without at least mentioning *agents*. Simply stated an agent can be viewed as a self-contained concurrently executing software process, that encapsulates some state and is able to communicate with other agents via message passing [25].

The provision of navigation support within virtual worlds is not new. Significant work has been undertaken of late [4,24] in the design of intelligent support for VE navigation. The work of Dijk et al. [4] has specifically sought to make provision for navigation assistance in the form of an embodied agent within the VE. However, this work merely encodes some general navigation heuristics and does not dynamically learn or deduce the navigational class of the individual user based upon trajectory analysis. Furthermore, the agents commissioned in this work [16] merely embrace a weak notion of agency and are in contrast to our approach

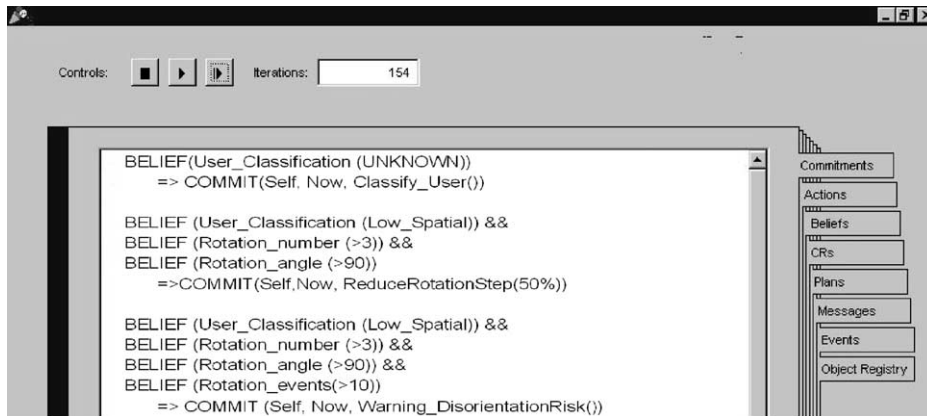


Fig. 5. The mental state of the navigation support agent.

where strong BDI agents are utilised in the provision of navigational support.

We have developed a navigational agent that currently is not embodied nor depicted as an avatar within the VE, although this is not precluded in future work. The nature and manner of our agent rather seeks to be least intrusive and indeed in many cases the user ought to be unaware of its very existence. The navigation support agent acts in a proactive nature anticipating the navigational needs of the user and taking actions accordingly. The agent is realised via Agent Factory a rapid prototyping environment for agent design and delivery [2,17]. Central to our agents is a rich mental state comprising of an aggregation of a set of beliefs, a set of commitments to future directed actions and a set of commitment rules that drive the adoption of commitments.

Fig. 5 depicts the mental state of our navigation support agent at a given instance in time. The mental state of an agent continually evolves with respect to time as the agent perceives new environmental events and deduces new observations about the user behaviour. Each will result in belief update and the addition and/or dropping of beliefs. In turn the beliefs drive commitment rule activation and in turn commitment adoption and ultimately the activation of actions that assist in user navigation.

Considering Fig. 5 specifically, here we can see a set of commitment rules. The first quite simply states that if the user classification is unknown then the agent ought to commit to performing a classification. In re-

ality this results in a request to a classification agent, which will when able to do so return the classification via a communication. Inter-agent communication takes the form of speech acts and specifically our navigation support agent would receive an inform speech act as to the user category. An agent preceptor would recognise this communication and accordingly adopt a belief that the current user was of that particular category. The second and third commitment rules concern navigational interventions, which the agent is empowered to perform. In the case of the former if the user is of low spatial ability and there are a large number of excessive rotations then the system would implicitly degrade the responsiveness of the rotation step thereby assisting the user in exploration. This is a case in point where the user may remain unaware of the agent intervention. Commitment rule 3 in contrast alerts the user with a warning where an excessive amount of pronounced rotation events occur. In both cases the occurrences of these key navigational events would be logged by the listener agent and communicated to the navigational support agent. These rules are merely illustrative but demonstrate how having identified the user category relatively simplistic inferences can give the illusion of more sophisticated navigational support.

Navigational support however is but one aspect of intelligent support. We strive to provide dynamic scene reconstruction whereby users, who are patently lost or proving unable to accomplish retrieval tasks, are assisted. This could take the form of bringing the desired location or resource closer to the foreground within



the scene graph. Alternatively the user could be re-located adjacent to key way-finding objects or landmarks that the navigational support agent knows the user will recognise. We have described the process of dynamic 3D world reconstruction elsewhere and the specific mechanisms by which this can be achieved [8].

## 6. Conclusions

This study shows that an ANN could be successfully employed in modelling spatial behaviour in VEs, in terms of classifying users' motion trajectories performed on-line. Based on this classification, each new user can be associated with one of the clusters, and accordingly identified as employing efficient or inefficient navigational strategies.

The SOM and LVQ analysis led to the identification of five user trajectory clusters within the same VE. The accuracy of classification is above 85% which is a significant outcome given the relatively limited size of our training and testing sets. Within each cluster, trajectories share common features. Some of them were already identified while the others require further analysis. A future direction will be to extract the quantitative rules governing the clusters and to express them in a symbolic manner. The study findings could provide insights in understanding what do the efficient and inefficient strategies mean, by interpreting them through theoretical aspects of spatial cognition described by environmental psychology. Moreover, the study indicates that using neural networks as a tool in studying navigation can be beneficial for user modelling in the area of spatial knowledge acquisition. In permitting a comparative analysis between efficient and inefficient navigational strategies, this methodology could suggest how VEs might be better designed. Based on these results, further work will focus on assisting new users to improve their spatial abilities in exploring a new VE. After a period of navigation, SOM would be able to integrate the online trajectory within the appropriate cluster. If the user's trajectory history matches, for example, Cluster 2, the system will assist the user in his/her further exploration.

We have described our on-going work in the delivery of a navigational support agent, which utilises user categorisation together with observed navigational behaviour to implicitly make interventions and assist the

user in the navigation process. Furthermore we consider how real-time dynamic reconstruction of the VE could assist the user in their tasks.

## References

- [1] R. Amant, M. Riedl, A practical perception substrate for cognitive modelling in HCI, *Int. J. Human Comput. Stud.* 55 (1) (2001) 15–39.
- [2] R. Collier, G. O'Hare, T. Lowen, C. Rooney, Beyond prototyping in the factory of the agents, in: *Proceedings of the Third Central and Eastern European Conference on Multi-agent Systems (CEEMAS'03)*, Lecture Notes in Computer Science, vol. 2691, Springer, Berlin, 2003, pp. 383–393.
- [3] T. Delahunty, ECHOES: a cohabited virtual training environment, M.Sc. Thesis, Department of Computer Science, University College Dublin, 2001.
- [4] B.V. Dijk, R.O.D. Akker, A. Nijholt, J. Zwiers, Navigation assistance in virtual worlds, *Inform. Sci. Spec. Ser. Commun. Inform.* 6 (2003) 115–125.
- [5] C. Freksa, C. Habel, K. Wender, Spatial cognition, an interdisciplinary approach to representing and processing spatial knowledge, in: *Lecture Notes in Computer Science*, vol. 1404, Springer-Verlag, Berlin, 1998.
- [6] J. Gabbard, D. Hix, Taxonomy of usability characteristics in virtual environments, Virginia Polytechnic Institute, Blacksburg, 1997.
- [7] W. Grimson, C. Stauffer, L. Lee, R. Romano, Using adaptive tracking to classify and monitor activities in a site, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998, pp. 22–31.
- [8] T. Guinan, G. O'Hare, N. Doikov, ENTER: the personalisation and contextualisation of three-dimensional worlds, in: *Proceedings of the Eighth Euromicro Workshop on Parallel and Distributed Processing*, IEEE Computer Society Press, 2000, pp. 142–149.
- [9] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, New Jersey, 1994.
- [10] S. Kaski, Data exploration using self-organizing maps, Ph.D. Thesis, Helsinki University of Technology, 1997.
- [11] T. Kohonen, Self-organizing maps, in: *Springer Series in Information Sciences*, vol. 30, Springer-Verlag, Berlin, 2001.
- [12] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, SOM PAK: the self-organizing map program package, Technical Report No. A31, Helsinki University of Technology, Laboratory of Computer and Information Science, 1996.
- [13] T. Kohonen, J. Hynninen, J. Kangas, J. Laaksonen, K. Torkkola, LVQ PAK: the learning vector quantization program package, Technical Report No. 30, Helsinki University of Technology, Laboratory of Computer and Information Science, 1995.
- [14] H.V. Lint, S. Hoogendoorn, H.V. Zuylen, Freeway travel time prediction with state-space neural networks, Technical Report No. 02-2797, Transportation Research Board, 2002.
- [15] M. Lorr, *Cluster Analysis for Social Scientists*, Jossey-Bass Publishers, San Francisco, 1983.

- [16] A. Nijholt, J. Zwiers, B.V. Dijk, Maps, agents and dialogue for exploring a virtual world, in: *Proceedings of the Fifth World Multiconference on Systemics, Cybernetics and Informatics (SCI 2001)*, 2001, pp. 94–99.
- [17] G. O'Hare, S. Abbas, Commitment manipulation within agent factory, in: *Proceedings of the Decentralised Intelligent and Multi-agent Systems, DIMAS'95*, 1995.
- [18] G. O'Hare, K. Sewell, A. Murphy, T. Delahunty, An agent based approach to managing collaborative work within both a virtual environment and virtual community, in: *Proceedings of the Workshop on Intelligent Agents for Computer Supported Cooperative Work: Technologies and Risks*, 2000.
- [19] G. O'Hare, K. Sewell, A. Murphy, T. Delahunty, An immerse training experience, in: P. Brusilovsky, O. Stock, C. Strapparava (Eds.), *Lecture Notes in Computer Science*, vol. 1892, Springer-Verlag, Berlin, 2000, pp. 179–188.
- [20] J. Owens, A. Hunter, Application of the self-organizing map to trajectory classification, in: *Proceedings of the Third IEEE Workshop on Visual Surveillance*, 2000, pp. 77–83.
- [21] C. Sas, G. O'Hare, Presence and individual differences in virtual environment: usability study, in: H., Sharp, P., Chalk, J., LePeuple, J., Rosbottom, (Eds.), *Proceedings of the 16th British HCI Conference*, vol. 2, British Computer Society, London, 2002, pp. 50–53.
- [22] C. Sas, G. O'Hare, R. Reilly, A connectionist approach to modelling navigation: trajectory self-organization and prediction, in: N. Carbonell, C. Stephanidis (Eds.), *Proceedings of the Seventh ERCIM Workshop, User Interfaces for All*, 2002, pp. 111–116.
- [23] C. Sas, G. O'Hare, R. Reilly, On-line trajectory classification, in: *Proceedings of the International Conference on Computational Science, Workshop on Scientific Visualisation and Human–Machine Interaction in a Problem-solving Environment*, *Lecture Notes in Computer Science*, vol. 2659, Springer-Verlag, Berlin, 2003, pp. 1035–1044.
- [24] V. Volbracht, G. Domik, Developing effective navigation techniques in virtual 3D environments, in: *Proceedings of the Eurographics Workshop*, Springer-Verlag, Berlin, 2000, pp. 55–64.
- [25] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, *Knowledge Eng. Rev.* 10 (2) (1995) 115–152.



**Corina Sas** obtained her PhD from University College Dublin in 2004. She received her bachelor degrees in Psychology and Computer Science from Lucian Blaga University in Sibiu and her MA in Industrial Psychology from Babes-Bolyai University in Cluj-Napoca, Romania. Her research interests include Spatial Cognition, User Modelling and Presence. She has published in journals and international conferences in these areas.



**Gregory O'Hare** is the Head of Department of Computer Science at University College Dublin (UCD). Prior to this he was a member of faculty at the University of Manchester Institute of Science and Technology (UMIST). He is Director of the PRISM (Practice and Research in Intelligent Systems and Media) Laboratory within the Department of Computer Science. His research focuses upon Multi-agent Systems (MAS) and Mobile and Ubiquitous computing. He

has published some 140 journals and conference papers in these areas together with two textbooks. Gregory has secured some 2.5 million Euro research funding for research conducted in the PRISM laboratory and has acted as consultant for many national and international companies and organizations.



**Ronan Reilly** is Professor of Computer Science at NUI Maynooth. He has a background in the areas of psychology and computer science. His main research interests are in the computational modelling of the brain processes that underlie language and vision, with a specific interest in the modelling of reading. Modular system for on-line trajectory classification.